

Курс лекций “From Junior To Senior”

Курс лекций по программированию кросс-платформенных риаптайм приложений на примере разработки небольшого движка и игры. В процессе разработки будут рассматриваться большое количество косвенных вопросов связанных с программированием и поддержкой масштабных систем.

Курс рассчитан на людей умеющих программировать и способных самостоятельно писать приложения под Windows. Будут рассматриваться языки: C++ и STL, C# (платформа .Net и Mono), Php и SQL, ObjectiveC, JavaScript, Java, Lua, MaxScript.

А также API: OpenGL и язык шейдеров GLSL, OpenAL, Cocoa, WinAPI, XWindow, Windows Presentation Foundation.

Будут рассказывать тимлиды и сотрудники компаний: ООО “ДиПроСофт” и НПЦ “Спектр”. Некоторые темы будут рассказывать представители компаний: ООО “Кибертайм Систем”, ООО “Яндекс” и возможно другие специально приглашенные гости.

Лекции будут проходить еженедельно по четвергам в 16:00 по МСК в г. Екатеринбурге, по адресу Ленина 10. Далее будут выкладываться на канале YouTube <http://www.youtube.com/user/megadrone86> .

Страница лекций, где выложены дополнительные материалы

[Ссылка на Youtube канал](#)

SVN репозиторий с исходными кодами проекта

<http://code.google.com/p/jutos/>

Тезисный план лекции и корректировки темы будут производиться за неделю до предстоящей лекции в этом документе. Сам план лекций не окончательный и будет корректироваться.

1. Введение в курс. Системы контроля версий. Процесс компиляции и сборки бинарного кода. Основы современной трехмерной компьютерной графики.
 - a. На кого рассчитан курс. Конечная его цель. Формат преподнесения информации.
 - i. Необходимость проведения данного курса. Озвучить целевую аудиторию.
 - ii. Рассказать о направлении лекций, чтобы потенциальная аудитория смогла сформировать свои ожидания.
 - iii. Принцип построения лекций и практикумов.
 - b. Все этапы компиляции исходных кодов в бинарные файлы.
 - i. Системы контроля версий. SVN.
 - ii. Сборка в исполняемого файла приложения (C++):
 1. Обработка исходного кода препроцессором
 - a. директивы препроцессора
 - b. параметры компилятора, управляющие препроцессором
 - c. основные приемы использования препроцессора
 2. Компиляция в объектные файлы
 - a. Что такое объектные файлы
 - b. Процесс компиляции

- c. Отличия Windows и Unix
 - d. управляющие параметры компилятора
 - 3. Аспекты языка, связанные с компоновкой
 - a. ключевые слова `extern` и `static`
 - b. декорирование имен
 - 4. Стадия компоновки (linking)
 - a. Типы продукта компоновщика: исполняемый файл, статическая библиотека, динамическая библиотека
 - b. Процесс связывания
 - c. Управляющие параметры компоновщика
 - d. Динамическая загрузка библиотек
 - iii. Сборка в промежуточный байт-код (C#, Java):
 - 1. Что такое байт-код, Just In Time (JIT) компиляция
 - 2. Отсутствие явной стадии компоновки
 - 3. Особенности Java
 - 4. Особенности C#
 - c. Как работают современные ускорители трехмерной графики. Фиксированный и программируемый конвейер. Основы программирования 3D графики, API DirectX и семейство OpenGL.
 - i. Фундаментальные принципы построения 3D графики. Проецирование геометрии, растеризация, текстурирование.
 - ii. Принцип работы видеокарты, отличие CPU от GPU, графические конвейер, что такое шейдер.
 - iii. API Direct3D и OpenGL, плюсы и минусы.
- 2. Разработка концепции игры, от идеи к концепт-документу и дизайн-документу.
 - a. Чему мы хотим научить.
 - b. Что такое геймдизайн и зачем он нужен.
 - c. Что такое плохо и что такое хорошо? От общего к частному.
 - d. Рождение идеи. Управление идеями.
 - e. Рождение геймплея.
 - f. Сеттинг и стилистика.
 - g. Вопросы перед написанием концепта.
 - h. Пишем концепт. Структура концепт-документа.
 - i. Прототипирование.
 - j. Из чего складывается дизайн документ.
 - k. Основные проблемы начинающего гейм девелопера.
- 3. Разбиение проекта на модули и постановка задач, а также планирование работ.
 - a. Что за проект? Какие этапы работ у нас будут.
 - b. Популярные способы планирования и тестирование.
 - c. Управление функциональными областями на всех стадиях жизненного цикла проекта.
 - d. Решение сложных ситуаций. Дерево существующей реальности.
- 4. Особенности архитектуры приложений реального времени. Проектирование архитектуры приложения (игра, редактор, конвертеры(экспортеры) серверная часть). Уровни абстракции. Особенности проектирования кросс-платформерных систем.
 - a. Платформы? Движок? С какой стороны подступиться?

б. Проектируем кросс-платформерный движок.

5. Нотация кода и кодстайл. Проектирование и разработка кросс-платформерного движка (создание окна, взаимодействие с оконной системой). Реализация ядра в среде Windows.
6. Доработка движка, инициализация OpenGL, подсистема пользовательского ввода.
7. Портируемся под Linux.
8. Портируем под MacOS X и iOS. Cocoa (Cocoa Mobile) и Carbon.
9. Портируем на мобильные устройства Android.
10. Работа с OpenGL. Различия семейства API OpenGL (OpenGL 1.x-2.x, OpenGL 3.x-4.x, OpenGL ES 1.x-2.x, WebGL). Работа с матрицами, пространственные преобразования. Использование шейдеров. Вывод геометрии и текстурирование.
11. Проектирование и разработка менеджера ресурсов. Файловая подсистема. Работа со звуком и музыкой на примере OpenAL.
12. Проектирование форматов хранения данных (паки, текстуры, шрифты, меши). И разработка инструментов экспорта (MaxScript, консольные конвертеры). Реализация поддержки форматов в движке.
13. Проектирование и разработка графа сцены. Оперирование объектами сцены рендера.
14. Создание игровых объектов. Разработка интерфейсов взаимодействия со скриптовым языком на примере привязки к Lua.
15. Создание редактора уровней (C#, WPF) игровых сцен. Реализация формата сцен.
16. Создания редактора (C#, WPF) и формата эффектов и внедрение его в движок.
17. Разработка игровой графики. От скетча до 3D модели. Дизайн уровней.
18. Разработка игровой логики и AI (Lua).
19. Проектирование клиент-серверного взаимодействия и сетевой игры. Реализация серверной части (Java сервер) и Web-портала (php).
20. Разработка клиентской части, работа с сокетами. Проблема лага и интерполяция, как решение. Взаимодействие по протоколами UDP и HTTP. Проектирование и разработка своего протокола на основе UDP.
21. Подготовка приложения для размещения в Apple App Store, Apple Mac App Store, Ubuntu Software Center, Android Market.
22. Портирование на HTML5. Реализация движка на JavaScript и WebGL.
23. Портирование игры, проблемы и решения.
24. Платформа Windows Phone 7. Вопросы портирования.
25. Заключительная лекция. Подведение итогов, анализ сложностей выявленных в процессе разработки. Что делать дальше?